JURNAL GAUSSIAN, Volume 1, Nomor 1, Tahun 2012, Halaman 21-30
Online di: http://ejournal-s1.undip.ac.id/index.php/gaussian

SIMULASI STOKASTIK MENGGUNAKAN ALGORITMA *GIBBS SAMPLING*

Anifa[1], Moch. Abdul Mukid[2], Agus Rusgiyono[3]
[1]Mahasiswa Jurusan Statistika FSM Universitas Diponegoro
[2,3]Staf Pengajar Jurusan Statistika FSM UNDIP

ABSTRACT

One way to get a random sample is using simulation. Simulation can be done directly or indirectly. Markov Chain Monte Carlo (MCMC) is an indirectly simulation method. MCMC method has some algorithms. In this thesis only discussed about Gibbs Sampling algorithm. Gibbs Sampling is introduced by Geman and Geman at 1984. This algorithm can be used if the conditional distribution of the target distribution is known. It has applied on two casses, these are generation of bivariate normal random data and parameters estimation using Bayesian method. The data used in this research are the death of pulmonary tuberculosis in ASEAN in 2007. The results obtained are $\hat{\mu} = 26{,}41$ and $\hat{\sigma} = 91{,}42$ with standard error for $\hat{\mu} = 0{,}78$ and $\hat{\sigma} = 0{,}32$.

Keywords: Simulation, indirect simulation, generation of random data, Markov Chain Monte Carlo, MCMC, Gibbs Sampling

1. PENDAHULUAN

Dalam berbagai bidang, penentuan suatu keputusan akan menjadi bagian terpenting. Inilah salah satu peran yang harus dijalankan oleh seorang statistisi yaitu dapat mengambil keputusan terbaik pada berbagai situasi dan kondisi. Selain mempertimbangkan faktor-faktor yang berkaitan, dalam menentukan keputusan juga penting untuk mempertimbangkan hasil perhitungan statistik yang ada. Dalam pehitungan tersebut akan membutuhkan suatu data yang merupakan informasi dari masalah tersebut. Data yang digunakan dapat berupa data primer, data sekunder, maupun data simulasi (data yang dibangkitkan). Oleh karena itu diperlukan suatu simulasi untuk membangkitkan sampel random. Mengutip tulisan I Made Tirta dalam buku "Pengantar Metode Simulasi Statistika dalam Aplikasi R dan S$^+$", simulasi adalah teknik untuk membuat konstruksi model matematika untuk suatu proses atau situasi dalam rangka menduga secara karakteristik atau menyelesaikan masalah berkaitan dengannya dengan menggunakan model yang diajukan. Jadi, dibutuhkan media komputer sebagai alat untuk melakukan simulasi pembangkitan sampel random tersebut.

Kasus-kasus dengan distribusi peluang yang memiliki bentuk umum dan dikenal seperti distribusi normal, beta, gamma, dan t dapat diselesaikan dengan cara simulasi langsung. Namun, untuk kasus yang memiliki bentuk distribusi peluang yang belum dikenal, penyelesaiannya dilakukan dengan cara simulasi tidak langsung. Sampai dengan sekarang, metode simulasi tak langsung yang sering digunakan adalah metode *Markov Chain Monte Carlo* atau disingkat MCMC.

Nama Monte Carlo diambil dari nama sebuah kota di Monaco yang terkenal sebagai pusat kasino. Secara sistematik metode Monte Carlo mulai berkembang tahun 1944. Namun, sebelumnya pada tahun 1931 Kolmogorov menunjukkan hubungan antara proses stokastik Markov dengan persamaan differensial. Tahun 1908 seorang mahasiswa,W.S. Gosset menggunakan percobaan untuk membantunya menemukan distribusi koefisien korelasi. Pada tahun yang bersamaan ada mahasiswa menggunakan metode sampling untuk memantapkan keyakinannya pada distribusi yang disebutnya distribusi t.

Metode MCMC itu sendiri memiliki 2 algoritma yang telah populer yaitu algoritma *Metropolis-Hastings* dan algoritma *Gibbs Sampling*. Kedua algoritma tersebut memiliki syarat penggunaan masing-masing. Dalam tulisan ini akan dijelaskan mengenai penggunaan algoritma *Gibbs Sampling* dalam membangkitkan sampel random dari distribusi target tertentu. Dalam contoh penerapan dikaji pula peggunaan *Gibbs Sampling* untuk estimasi parameter dengan metode Bayes.

2. TINJAUAN PUSTAKA

Tinjauan pustaka yang digunakan dalam tulisan ini adalah sebagai berikut:

2.1. Distribusi Bersama Variabel Random Kontinu

Fungsi densitas probabilitas bersama dari suatu variabel random kontinu $X = (X_1, X_2, ..., X_k)$ berdimensi-k didefinisikan sebagai:

$$f(x_1, x_2, ..., x_k) = P[X_1 = x_1, X_2 = x_2, ..., X_k = x_k]$$

untuk setiap nilai x yang mungkin. Bila suatu variabel random kontinu (X_1, X_2)memiliki pdf bersama $f(x_1, x_2)$maka pdf marginal untuk X_1,dan X_2 adalah:

$$f_1(x_1) = \int_{-\infty}^{\infty} f(x_1, x_2) dx_2$$
$$f_2(x_2) = \int_{-\infty}^{\infty} f(x_1, x_2) dx_1$$

Fungsi distribusi kumulatif bersama dari suatu variabel random kontinu $X = (X_1, X_2, ..., X_k)$ berdimensi-k didefinisikan sebagai:

$$F(x_1, x_2, ..., x_k) = P[X_1 \le x_1, X_2 \le x_2, ..., X_k \le x_k]$$

$$F(x_1, x_2, ..., x_k) = \int_{-\infty}^{x_k} ... \int_{-\infty}^{x_1} f(t_1 ... t_k) dt_1 ... dt_k, \qquad \text{untuk setiap x} = (x_1, ..., x_k).$$

(Bain dan Engelhardt, 1992)

2.2. Distribusi Bersyarat

Distribusi bersyarat sering juga disebut sebagai distribusi kondisional yaitu suatu distribusi dari sebuah kejadian, misalkan A, dengan syarat bahwa suatu kejadian lain, misalkan B, telah terjadi.

Suatu variabel random (X_1, X_2)dengan pdf bersama $f(x_1, x_2)$ memiliki pdf bersyarat dari X_2 dengan syarat $X_1 = x_1$ didefinisikan:

$$f(x_2|x_1) = \frac{f(x_1, x_2)}{f(x_1)} \tag{1}$$

Sedangkan pdf bersyarat dari X_1 dengan syarat $X_2 = x_2$ didefinisikan:

$$f(x_1|x_2) = \frac{f(x_1, x_2)}{f(x_2)} \tag{2}$$

(Bain dan Engelhardt, 1992)

2.3. Fungsi Likelihood

Fungsi likelihood adalah fungsi densitas bersama dari n variabel random $X_1, X_2, ..., X_n$ dan dinyatakan dalam bentuk $f(x_1, x_2, ..., x_n|\theta)$. Jika $x_1, x_2, ..., x_n$ tetap, maka fungsi likelihood adalah fungsi dari parameter θ dan dinotasikan dengan $L(\theta)$. Jika $x_1, x_2, ..., x_n$ menyatakan suatu sampel random dari $f(x|\theta)$, maka:

$$L(\theta) = f(x_1|\theta) f(x_2|\theta) ... f(x_n|\theta) = \prod_{i=1}^{n} f(x_i|\theta) \tag{3}$$

(Bain dan Engelhardt, 1992)

2.4. Distribusi Prior

Di dalam analisis Bayesian, ketika suatu populasi mengikuti distribusi tertentu dengan suatu parameter didalamnya, misal θ, maka dimungkinkan bahwa parameter θ itu sendiri juga mengikuti suatu distribusi probabilitas tertentu, yang disebut sebagai distribusi prior. Distribusi prior seringkali dituliskan dengan notasi $f(\theta)$, sehingga dapat dituliskan $\theta \sim f(\theta)$. Terkadang ditemui masalah dalam memilih fungsi densitas dari prior $f(\theta)$ dan dalam beberapa kasus θ dapat diasumsikan sebagai suatu variabel random, baik variabel random diskrit ataupun variabel random kontinu.

(Bain dan Engelhardt, 1992)

2.5. Distribusi Posterior

Distribusi posterior adalah fungsi densitas bersyarat θ jika nilai observasi x diketahui dan dapat dituliskan sebagai berikut:

$$f(\theta|x_i) = \frac{f(\theta, x_i)}{f(x_i)} \tag{4}$$

Apabila θ kontinu, distribusi prior dan posterior θ dapat disajikan dengan fungsi kepadatan. Fungsi kepadatan bersyarat satu variabel random jika diketahui nilai variabel random kedua hanyalah fungsi kepadatan bersama dua variabel random itu dibagi dengan fungsi kepadatan marginal variabel random kedua. Tetapi fungsi kepadatan bersama $f(\theta, x_i)$ dan fungsi kepadatan margial $f(x_i)$ pada umumnya tidak diketahui, hanya distribusi prior dan fungsi likelihood yang biasanya dinyatakan.

Fungsi kepadatan bersama dan marginal yang diperlukan dapat ditulis dalam bentuk distribusi prior dan fungsi likelihood,

$$f(\theta, x_i) = f(\theta) f(x_i|\theta)$$

dimana $f(x_i|\theta)$ merupakan fungsi likelihood dan $f(\theta)$ merupakan distribusi prior. Selanjutnya diketahui bahwa

$$f(x_i) = \int_{-\infty}^{\infty} f(\theta, x_i) d\theta = \int_{-\infty}^{\infty} f(\theta) f(x_i|\theta) d\theta$$

Sehingga fungsi kepadatan posterior untuk variabel random kontinu dapat ditulis sebagai:

$$f(\theta|x_i) = \frac{f(\theta) f(x_i|\theta)}{\int_{-\infty}^{\infty} f(\theta) f(x_i|\theta) d\theta} \tag{5}$$

(Soejoeti dan Soebanar, 1988)

2.6. Metode Bayesian

Pada metode Bayesian, inferensi didasarkan pada distribusi posterior $f_{\theta|x}(\theta)$, yang dapat juga dituliskan sebagai $f(\theta|x_1, \dots, x_n)$. Dari persamaan (5) maka:

$$f(\theta|x_1,\dots,x_n) = \frac{f(x_1,\dots,x_n|\theta) f(\theta)}{\int f(x_1,\dots,x_n|\theta) f(\theta) d\theta}$$

$$f(\theta|x_1,\dots,x_n) = \frac{f(x_1,\dots,x_n|\theta) f(\theta)}{m(x_1,\dots,x_n)}$$

Faktor penyebut, yaitu m$(x_1, \dots, x_n)$ merupakan fungsi likelihood marginal dari data, sehingga dapat dirumuskan bahwa:

$$\text{m}(x_1, \dots, x_n) = \int f(x_1, \dots, x_n|\theta) f(\theta) d\theta$$

Dalam metode Bayesian dikenal suatu faktor kesebandingan untuk menentukan distribusi posterior $f(\theta|x_1,\dots,x_n)$, yaitu:

$$f(\theta|x_1,\dots,x_n) \propto f(x_1,\dots,x_n|\theta) f(\theta)$$

Lambang $\propto$ menyatakan sifat proporsional atau sebanding. Pada perspektif Bayesian fungsi likelihood merupakan fungsi dari θ pada data $x_1, \dots, x_n$ sehingga mengakibatkan elemen –

elemen likelihood yang tidak mengandung fungsi θ menjadi bagian dari kesebandingan. Dengan kata lain melalui sifat kesebandingan diperoleh bahwa densitas posterior hanya mengandung fungsi yang memuat θ.

Penduga Bayes untuk θ diperoleh melalui nilai harapan dari $\left(\theta|x_1, x_2, ..., x_n\right)$, dapat ditulis:

$$\hat{\theta} = E\left(\theta|x_1, x_2, ..., x_n\right).$$

(Congdon, 2003)

2.7. Metode *Markov Chain Monte Carlo* (MCMC)

Dewasa ini metode *Markov Chain Monte Carlo*(MCMC) telah banyak diaplikasikan di berbagai bidang untuk menyelesaikan bermacam–macam permasalahan, khususnya yang terkait dengan inferensi Bayesian. yang berkaitan dengan persoalan mendapatkan suatu distribusi posterior dan juga distribusi prior pada beberapa studi kasus. Metode MCMC dapat digunakan baik untuk kasus univariat maupun multivariat. Metode ini memiliki 2 algoritma yang sering digunakan yaitu algoritma *Metropolis-Hastings* dan algoritma *Gibbs Sampling*. Pada tulisan ini hanya akan membahas mengenai algoritma *Gibbs Sampling*.

(Walsh, 2004)

2.7.1 Algoritma *Gibbs Sampling*

Gibbs Sampling diperkenalkan oleh Geman dan Geman (1984). Algoritma ini merupakan kasus khusus dari komponen tunggal algoritma Metropolis-Hastings yang menggunakan densitas proposal $q(x'|x^{(t)})$, yaitu distribusi target bersyarat penuh $f(x_j|\mathbf{x}_{\setminus j})$, dimana $\mathbf{x}_{\setminus j} = (x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_p)^T$. Distribusi proposal seperti ini menghasilkan peluang penerimaan $\alpha = 1$, dan oleh karena itu perpindahan yang diusulkan diterima untuk semua iterasi. Meskipun *Gibbs Sampling* merupakan kasus khusus dari algoritma Metropolis-Hasting, biasanya disebut juga sebagai teknik simulasi yang terpisah karena kepopuleran dan kemudahannya. Salah satu keuntungan dari *Gibbs sampling* tersebut yaitu, pada setiap langkah, nilai random harus dibangkitkan dari distribusi dimensi tunggal yang mana alat-alat komputasi yang tersedia beragam jenisnya. Seringkali, distribusi bersyarat ini memiliki bentuk yang diketahui, sehingga sejumlah nilai random dapat disimulasi dengan mudah menggunakan fungsi standar pada software statistik dan komputasi. *Gibbs sampling* selalu bergerak ke nilai-nilai baru dan yang paling penting adalah tidak memerlukan spesifikasi dari distribusi-distribusi proposal. Pada sisi lain, ini dapat tidak berguna ketika ruang parameter rumit atau parameter-parameter sangat berkorelasi.

Algoritma *Gibbs Sampling* dapat diringkas dengan langkah-langkah sebagai berikut:

Misalkan $\mathbf{x} = (x_1, x_2, ..., x_p)$

1. Menentukan nilai awal $\boldsymbol{x}^{(0)}$
2. Untuk $t = 1, \dots, T$ ulangi langkah-langkah berikut
   a. Menentukan $\boldsymbol{x} = \boldsymbol{x}^{(t-1)}$
   b. Untuk $j = 1, \dots, p$ perbaharui x_j dari $x_j \sim f(x_j\ |\mathbf{x}_{\setminus j})$. Proses lengkapnya sebagai berikut:

$x_1^{(t)}$ dari $f\left(x_1 \middle| x_2^{(t-1)}, x_3^{(t-1)}, ... x_p^{(t-1)}\right)$,

$x_2^{(t)}$ dari $f\left(x_2 \middle| x_1^{(t)}, x_3^{(t-1)}, ... x_p^{(t-1)}\right)$,

$x_3^{(t)}$ dari $f\left(x_3 \middle| x_1^{(t)}, x_2^{(t)}, x_4^{(t-1)} ... x_p^{(t-1)}\right)$

. . . . . .

$x_j^{(t)}$ dari $f\left(x_j \middle| x_1^{(t)}, x_2^{(t)}, ..., x_{j-1}^{(t)}, x_{j+1}^{(t-1)} ... x_p^{(t-1)}\right)$

. . . . . .

. . . . . .

$x_p^{(t)}$ dari $f\left(x_p \middle| x_1^{(t)}, x_2^{(t)}, \ldots x_{p-1}^{(t)}\right)$ (6)

c. Membentuk $x^{(t)}$ dan menyimpannya sebagai himpunan nilai-nilai yang dibangkitkan pada iterasi ke-$(t+1)$ dari algoritma

Membangkitkan nilai nilai dari $f\left(x_j \middle| \mathbf{x}_{\setminus j}\right) = f\left(x_j \middle| x_1^{(t)}, \ldots, x_{j-1}^{(t)}, x_{j+1}^{(t-1)}, \ldots, x_p^{(t-1)}\right)$ relatif mudah karena merupakan distribusi univariat dimana semua variabel-variabel kecuali x_j dipertahankan tetap pada nilai-nilai yang diberikannya.

(Ntzoufras, 2009)

2.7.2 Konvergensi Algoritma

Dibawah ini akan dijelaskan 3 metode yang sering digunakan dalam monitoring konvergensi.

1. *Trace Plot*

   Salah satu cara pendugaan *burn-in periode* adalah memeriksa *trace plot* nilai simulasi dari komponen atau beberapa fungsi lainnya dari **x** terhadap jumlah iterasi. *Trace plot* merupakan gambaran sebuah plot dari iterasi versus nilai yang telah dibangkitkan. *Trace plot* terutama sekali penting ketika algoritma MCMC dimulai dengan nilai-nilai parameter yang jauh dari pusat distribusi target. Pada kasus seperti itu, nilai-nilai simulasi dari **x** pada awal iterasi algoritma akan menyimpang dari daerah ruang parameter dimana distribusi target dipusatkan. Sebuah *trend* naik atau turun pada nilai parameter pada *trace plot* menunjukkan bahwa *burn-in period* belum tercapai. Jika tren-tren seperti ini muncul, maka penting untuk menghilangkan bagian awal dari rantai, karena nilai-nilai awal ini tidak menunjukkan perkiraan sampel yang benar dari distribusi target. Dengan kata lain, jika semua nilai-nilai berada dalam sebuah daerah tanpa keperiodikan yang kuat cenderung dapat diasumsikan konvergen.

2. Autokorelasi

   Untuk kedua algoritma MH dan *Gibbs sampling*, nilai simulasi x pada iterasi ke-$(t+1)$ bergantung pada nilai simulasi pada iterasi ke-t. Jika pada rantai terdapat korelasi yang kuat diantara nilai-nilai yang berurutan, maka kedua nilai berurutan tersebut memberikan informasi hanya secara marginal mengenai distribusi target dan bukan nilai dari sebuah simulasi tunggal. Korelasi yang kuat diantara iterasi yang berurutan menunjukkan bahwa algoritma masih berada pada daerah tertentu dari ruang parameter dan mungkin membutuhkan waktu yang lama untuk penyampelan dari keseluruhan daerah distribusi.

   Statistik yang umum digunakan untuk mengukur tingkat ketergantungan diantara pengambilan berurutan pada rantai adalah autokorelasi. Autokorelasi mengukur korelasi diantara kumpulan nilai-nilai simulasi $\{x_j^{(t)}\}$ dan $\{x_j^{(t+L)}\}$, dimana L merupakan jumlah lag dari iterasi terpisah pada dua kumpulan nilai-nilai. Untuk komponen tertentu, fungsi autokorelasi dapat dihitung sebagai fungsi nilai-nilai yang berbeda dari lag, L. Untuk komponen j, autokorelasi lag L dapat diduga dengan

$$r_{jL} = \frac{T'}{T'-L} \frac{\sum_{j=1}^{T'-L} \left(x_j - \bar{x}\right)\left(x_{j+L} - \bar{x}\right)}{\sum_{j=1}^{T'} \left(x_j - \bar{x}\right)^2}$$

   dimana $\bar{x}$ merupkan rata-rata dari nilai-nilai simulasi. Nilai autokorelasi untuk lag 1 akan hampir selalu menjadi positif untuk algoritma MH dan Gibbs sampling.

3. *Ergodic Mean Plot*

   Ergodic mean adalah istilah yang menunjukkan nilai mean sampai dengan *current iteration*. Plot antara iterasi dengan nilai mean disebut dengan *ergodic mean plot*. Jika setelah beberapa iterasi *ergodic mean* stabil, maka ini merupakan sebuah indikasi konvergensi dari algoritma telah tercapai.

(Ntzoufras, 2009)

2.7.3 Pendugaan Parameter

Pendugaan parameter dengan menggunakan metode MCMC biasanya digunakan untuk kasus-kasus inferensi Bayesian. Dengan menjalankan sebuah algoritma MCMC, nilai-nilai simulasi $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(T')}$ masing-masing terdistribusi secara kira-kira ke distribusi posterior $f\left(\theta \middle| x_i\right)$.

Penduga dari parameter θ diperoleh dari nilai rata-rata dari nilai-nilai sampel yang tersimulasi, yaitu:

$$\hat{\theta} = \frac{\sum_{t=1}^{T'} \theta^{(t)}}{T'} \tag{7}$$

Setelah didapatkan dugaan parameter, perhitungan penting lainnya pada analisis output adalah mengenai *standard error*. Untuk menghitung *standard error* dari estimasi ini dapat dilakukan dengan metode *batch means*. Metode *batch means* merupakan salah satu metode yang sederhana dan mudah diterapkan. Metode ini dilakukan dengan membagi lagi urutan nilai-nilai simulasi $\theta^{(1)},...,\theta^{(T')}$ menjadi b kelompok dengan setiap kelompoknya berukuran v, sehingga $T' = bv$. Untuk setiap kelompok dihitung rata-rata sampel, misal rata-rata kelompok sampel adalah $\hat{\theta}^1,...,\hat{\theta}^b$. Misalkan bahwa, ukuran kelompok v yang telah dipilih cukup besar sehingga autokorelasi (lag 1) pada rangkaian *batch means* kecil, katakan dibawah 0.1, maka estimasi *standard error* $\hat{\theta}$ dapat diduga dengan standard deviasi dari *batch means*, yaitu:

$$S_{\hat{\theta}}^{\ B} = \sqrt{\frac{\sum_{l=1}^{b} \left(\bar{\theta}^{\,l} - \bar{\theta}\right)^2}{(b-1)b}} \tag{8}$$

Standard error ini sangat berguna untuk menentukan ketelitian dari rata-rata posterior yang dihitung dalam simulasi yang dijalankan. Pada kejadian tersebut, jika *standard error* terlalu besar, maka algoritma MCMC sebaiknya dijalankan kembali menggunakan jumlah iterasi yang lebih besar.

(Johnson dan Albert, 1999)

3. HASIL DAN PEMBAHASAN

Pada tulisan ini diberikan dua contoh kasus, yaitu pembangkitan sampel random dari distribusi normal bivariat dan pendugaan parameter dengan metode Bayes.

3.1 Penerapan Gibbs Sampling untuk Pembangkitan Sampel Random pada Distribusi Normal Bivariat

Akan dibangkitkan sampel random dari distribusi normal bivariat yang fungsi densitasnya adalah sebagai berikut:

$$f(x_1, x_2) = \frac{1}{2\pi\sqrt{\sigma_{11}\sigma_{22}\left(1-\rho^2{}_{12}\right)}} \exp\left\{ -\frac{1}{2\left(1-\rho^2{}_{12}\right)} \left[\begin{array}{l} \left(\frac{x_1-\mu_1}{\sqrt{\sigma_{11}}}\right)^2 + \left(\frac{x_2-\mu_2}{\sqrt{\sigma_{22}}}\right)^2 \\ -2\rho_{12}\left(\frac{x_1-\mu_1}{\sqrt{\sigma_{11}}}\right)\left(\frac{x_2-\mu_2}{\sqrt{\sigma_{22}}}\right) \end{array} \right] \right\}$$

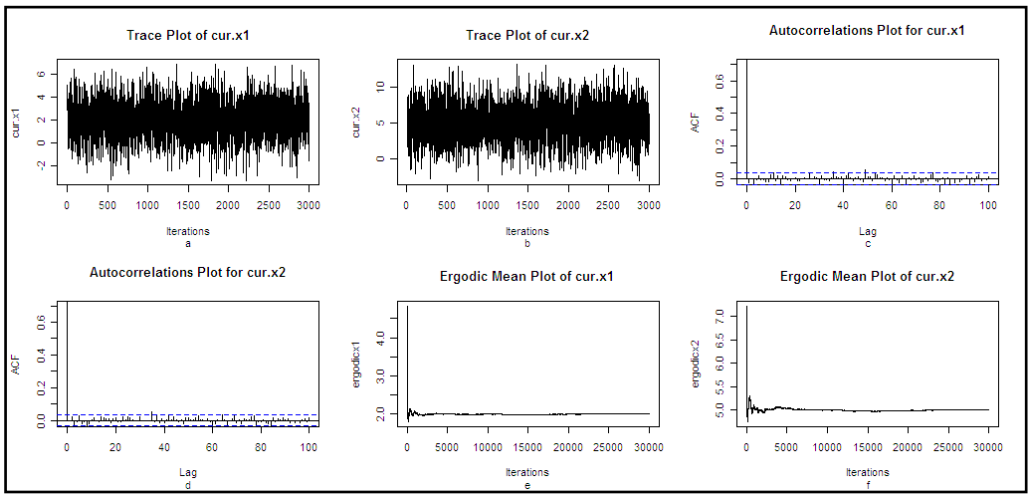
dengan $\tilde{X} = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}$, $\tilde{\mu} = \begin{pmatrix} 2 \\ 5 \end{pmatrix}$, dan $\tilde{\Sigma} = \begin{pmatrix} 2 & 2 \\ 2 & 4 \end{pmatrix}$

Dengan merujuk Johnson dan Wichern (2007) didapatkan distribusi bersyaratnya adalah

$$\left(X_1 \middle| X_2 = x_2\right) \sim N\left(\mu_1 + \frac{\sigma_{12}}{\sigma_{22}}\left(x_2 - \mu_2\right), \sigma_{11} - \frac{\sigma^2{}_{12}}{\sigma_{22}}\right) \tag{9}$$

$$\left(X_2 \middle| X_1 = x_1\right) \sim N\left(\mu_2 + \frac{\sigma_{12}}{\sigma_{11}}\left(x_1 - \mu_1\right), \sigma_{22} - \frac{\sigma^2{}_{12}}{\sigma_{11}}\right) \tag{10}$$

Dari distribusi bersyarat yang telah ditemukan yaitu pada persamaan (9) dan persamaan (10) maka pembangkitan sampel random yang akan dijalankan dengan *software* R 2.10.0. Pada kasus ini digunakan nilai awal $X_2 = 8$ dengan dilakukan iterasi sebanyak 30000 kali dan *lag sampling* = 10. Berikut output yang dihasilkan:



Gambar 1 (a) *Trace plot* dari nilai x_1 yang dibangkitkan, (b) *Trace plot* dari nilai x_2 yang dibangkitkan, (c) Plot autokorelasi nilai x_1 yang dibangkitkan, (d) Plot autokorelasi nilai x_2 yang dibangkitkan, (d) *Ergodic mean plot* nilai x_1 yang dibangkitkan, (f) *Ergodic mean plot* nilai x_2 yang dibangkitkan

Dari output pada Gambar 1 terlihat bahwa *trace plot* untuk nilai x_1 dan x_2 sudah tidak membentuk pola sehingga menunjukkan bahwa proses *burn-in* telah selesai. Pada gambar plot autokorelasi terlihat bahwa nilai-nilai autokorelasi pada lag pertama mendekati satu dan selanjutnya nila-nilainya terus berkurang menuju 0 sehingga dapat dikatakan bahwa pada rantai terdapat korelasi yang lemah. Korelasi yang lemah menunjukkan bahwa algoritma sudah berada di dalam daerah distribusi target. Gambar yang terakhir merupakan gambar *ergodic mean plot*, pada gambar tersebut keduanya sudah menunjukkan bahwa *ergodic mean* sudah stabil. Proses *Burn-in* untuk nilai x_1 selesai pada iterasi ke-20000 sedangkan untuk nilai x_2 *burn-in* selesai pada iterasi ke-25000, sehingga untuk kedua nilai tersebut diambil nilai *burn-in period* pada iterasi ke-25000. Dari ketiga metode yang digunakan telah membuktikan bahwa data yang dibangkitkan telah konvergen, yaitu data berasal dari distribusi targetnya.

Setelah konvergensi tercapai, data yang dibangkitkan diuji dengan uji Kolmogorov-Smirnov dan menghasilkan nilai p-value sebesar 0.8908 sehingga dapat dikatakan bahwa data yang dibangkitkan berasal dari distribusi normal bivariat.

3.2 Penerapan Gibbs Sampling untuk Pendugaan Parameter pada Kasus Bayesian

Pada contoh ini akan dilakukan pendugaan parameter dari populasi banyaknya kematian yang berhubungan dengan tuberkulosis paru per 100000 penduduk di negara-negara ASEAN tahun 2007. Pendugaan parameter dilakukan dengan menggunakan metode Bayes dan algoritma Gibbs sampling. Berdasarkan uji Kolmogorov-Smirnov dihasilkan bahwa sampel yang diperoleh berdistribusi normal atau $y_i \sim N(\mu, \sigma^2)$.

Distribusi prior yang digunakan adalah prior non konjugat, yaitu $\mu \sim N(\mu_0, \sigma^2{}_0)$ dan $\sigma^2 \sim IG(a_0, b_0)$ dengan parameter $\mu_0 = 0, \sigma_0 = 100,$ $a_0 = 0{,}002,$ dan $b_0 = 0{,}002$. Ini

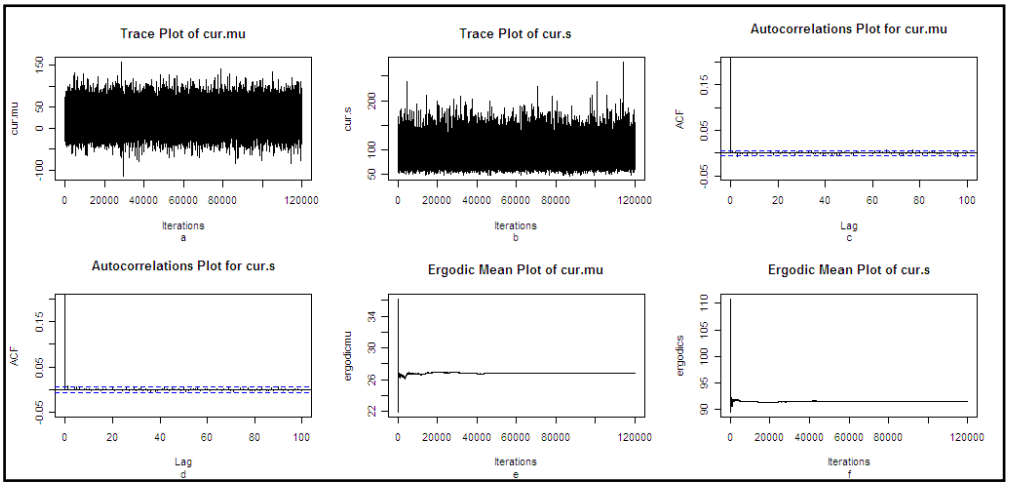
merupakan distribusi prior dengan informasi yang rendah. Ukuran sampel dan rata-ratanya adalah $n = 18, \bar{y} = 28{,}06$.

Dengan merujuk Ntzoufras (2009) diketahui bahwa distribusi bersyaratnya adalah

$$\left(\mu|\sigma^2\right) \sim N\left(w\bar{y} + (1-w)/\,\mu_0, w\frac{\sigma^2}{n}\right), \text{ dimana } w = \frac{\sigma^2{}_0}{\sigma^2/n + \sigma^2{}_0}$$

dan $$\left(\sigma^2|\mu\right) \sim IG\left(a_0 + \frac{n}{2}, b_0 + \frac{1}{2}\sum_{i=1}^{n}(y_i - \mu)^2\right)$$

Dari fungsi bersyarat yang telah ditemukan maka pembangkitan sampel random dijalankan menggunakan *software* R 2.10.0. Pada kasus ini dijalankan dengan iterasi sebanyak 120000 kali dengan *lag sampling* = 10. Berikut output yang dihasilkan:



Gambar 2 (a) *Trace plot* dari nilai μ yang dibangkitkan, (b) *Trace plot* dari nilai σ yang dibangkitkan, (c) Plot autokorelasi nilai μ yang dibangkitkan, (d) Plot autokorelasi nilai σ yang dibangkitkan, (d) *Ergodic mean plot* nilai μ yang dibangkitkan, (f) *Ergodic mean* plot nilai σ yang dibangkitkan

Dari output pada Gambar 3 terlihat bahwa *trace plot* untuk nilai μ dan σ sudah tidak membentuk pola sehingga menunjukkan bahwa proses *burn-in* telah selesai. Pada gambar plot autokorelasi terlihat bahwa nilai-nilai autokorelasi pada lag pertama mendekati satu dan selanjutnya nila-nilai terus berkurang menuju 0 sehingga dapat dikatakan bahwa pada rantai terdapat korelasi yang lemah. Korelasi yang lemah menunjukkan bahwa algoritma sudah berada di dalam daerah distribusi target. Gambar yang terakhir merupakan gambar *ergodic mean plot*, pada gambar tersebut keduanya sudah menunjukkan bahwa *ergodic mean* sudah stabil. Pada gambar (e) yaitu *ergodic mean plot* dari nilai rata-rata yang dibangkitkan terlihat stabil setelah iterasi ke 60000. Sedangkan untuk gambar (f) yaitu *ergodic mean plot* dari nilai simpangan baku yang dibangkitkan, terlihat stabil pada iterasi ke-80000. Keduanya sudah tidak membentuk pola, baik itu *trend* naik maupun turun. Sehingga dapat dikatakan bahwa *ergodic mean* sudah stabil. Batas iterasi ke-80000 itu merupakan nilai *burn-in period*. Jadi, hal ini telah mengindikasikan konvergensi algoritma dari sampel yang dibangkitkan.
Dari ketiga metode yang digunakan telah membuktikan bahwa data yang dibangkitkan sudah konvergen, yaitu data berasal dari distribusi targetnya.

Secara manual, pendugaan parameter dapat dilakukan dengan menggunakan rumus yang telah dijelaskan sebelumnya, yaitu pada persamaan (7). Hasil pendugaan yang didapatkan adalah $\hat{\mu} = 26{,}41$ dan $\hat{\sigma} = 91{,}42$. Perhitungan standard error untuk penduga parameter juga dilakukan secara manual yaitu dengan rumus pada persamaan (8). Hasil yang didapatkan adalah $S_{\hat{\mu}}{}^{B} = 0{,}78$ dan $S_{\hat{\sigma}}{}^{B} = 0{,}32$. Nilai *standard error* yang kecil mengindikasikan bahwa nilai yang yang dibangkitan memiliki tingkat kesalahan yang kecil, sehingga tidak perlu melakukan iterasi dalam jumlah yang lebih besar lagi.

4. KESIMPULAN

Salah satu metode simulasi tidak langsung yang sudah dikembangkan adalah metode Markov Chain Monte Carlo (MCMC). Metode ini memiliki berbagai macam algoritma yang salah satunya adalah algoritma Gibbs Sampling. Algoritma Gibbs Sampling dapat digunakan jika distribusi bersyarat tiap-tiap variabel dari distribusi target diketahui. Pada Gibbs Sampling semua simulasi adalah univariat dan akan menerima semua sampel hasil simulasi.

Pada pembahasan diberikan dua contoh penerapan metode Gibbs Sampling yaitu untuk pembangkitan sampel random pada distribusi normal bivariat dan pendugaan parameter pada kasus bayesian. Pada contoh pertama, dilakukan iterasi sebanyak 30000 kali dengan *burn-in period* pada iterasi ke-25000. Setelah konvergensi tercapai, data yang dibangkitkan diuji dengan uji Kolmogorov-Smirnov dan menghasilkan nilai p-value sebesar 0.8908 sehingga dapat dikatakan bahwa data yang dibangkitkan berasal dari distribusi normal bivariat. Sedangkan pada contoh kedua, dilakukan iterasi sebanyak 120000 kali dengan *burn-in period* pada iterasi ke-80000. Hasil pendugaan parameter untuk $\hat{\mu} = 26{,}41$ dan $\hat{\sigma} = 91{,}42$ dengan *standard error* $\hat{\mu} = 0{,}78$ dan $\hat{\sigma} = 0{,}32$.

DAFTAR PUSTAKA

Bain, L. J. dan Engelhardt, M. 1992. *Introduction to Probability and Mathematical Statistics Second Edition*. California: Duxbury Press.

Congdon, P. 2003. *Bayesian Statistical Modelling*. John Wiley: Chichester, UK.

Geman, S. dan Geman, D. 1984. *Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images*. IEE Transaction on Pattern Analysis and Machine Intelligence.

Johnson, R. A. dan Wichern, D. W. 2007. *Applied Multivariate Statistical Analysis. Sixth edition.* Prentice Hall International Inc: New Jersey.

Johnson, V. E. dan Albert, J. H. 1998. *Ordinal Data Modeling*. New York: Springer-Verlag Inc.

Ntzoufras, I. 2009. *Bayesian Modeling Using WinBUGS*. Greece: John Wiley.

Soejoeti, Z. dan Soebanar.1988. *Inferensi Bayesian*. Jakarta: Karunika Universitas Terbuka.

Tirta, I M. 2004. *Pengantar Metode Simulasi Statistika dengan Aplikasi R dan S^+*. Jember: FMIPA Universitas Jember.

Walsh, B. 2004. *Markov Chain Monte Carlo and Gibbs Sampling*. Lecture Notes for EEB 581,version 26 April 2004.